



Haz lo que hicimos: algunas conversiones en Java

Ahora practiquemos algunas conversiones en Java, para eso, sigue los pasos a continuación:

1) Crea una clase y escribe tu método principal, en el curso elegimos el nombre "TestConversion".

2) Escribe el siguiente código:

```
public class TestConversion {  
  
    public static void main(String[] args){  
        double salario = 1270.50;  
        int valor = salario;  
    }  
  
}
```

COPIA EL CÓDIGO

Este código no se compila, ya que estamos tratando de asignar una variable double dentro de una variable int, que no es posible en el lenguaje Java.

Sin embargo, recuerda que lo inverso es válido, es decir, podemos asignar un valor de tipo int dentro de una variable de tipo double.

3) Hay una manera de compilar este código. Para eso usaremos lo que llamamos casting, entonces, modifica tu código y agrega (int) antes de la variable de salario :

```
public class TestConversion {  
  
    public static void main(String[] args){  
        double salario = 1270.50;  
        int valor = (int) salario;  
    }  
  
}
```

[COPIA EL CÓDIGO](#)

Cuando hacemos el casting, le pedimos a Java que encuentre una manera de transformar el valor en el lado derecho de (int), en nuestro caso, la variable de salario, en su totalidad. De esa manera, Java transformará el double en int.

4) " Imprime " la variable de valor en la consola, para esto, agrega:

```
System.out.println (valor);
```

[COPIA EL CÓDIGO](#)

La salida será 1270.

Básicamente, para las variables primitivas, el casting no es más que hacer que esta conversión se realice cuando no se realiza automáticamente.

5) Por el momento, nuestra variable salario es del tipo double, ¿es esta la mejor opción? Prueba lo siguiente:

Escribe 3 variables de tipo double como se hace a continuación, la última variable (total) recibirá la suma de las otras dos, ten en cuenta:

```
public class TestConversion {
```

```
public static void main(String[] args){

    double salario = 1270.50;
    int valor = (int) salario;
    System.out.println(valor);

    double valor1 = 0.2;
    double valor2 = 0.1;
    double total = valor1 + valor2;

    System.out.println(total);

}
}
```

[COPIA EL CÓDIGO](#)

Imaginamos que nuestra salida sería 0.3, pero obtenemos un número completamente diferente.

Debido a esta inconsistencia, el double no sería la mejor opción para guardar salarios. Hay otras mejores alternativas que veremos en otros cursos.

6) También probaremos el tipo float, para esto escribe la variable puntoflotante tipo float:

```
public class TestConversion {

    public static void main(String[] args){

        float puntoFlotante = 3.14;    // aqui!

        double salario = 1270.50;
        int valor = (int) salario;
```

```
        System.out.println(valor);

        double valor1 = 0.2;
        double valor2 = 0.1;
        double total = valor1 + valor2;

        System.out.println(total);
    }

}
```

[COPIA EL CÓDIGO](#)

Por lo tanto, tendremos un error, ya que, para Java, 3.14 es un double. Este es el mismo problema que tuvimos al intentar almacenar un double dentro de un int. En el tipo double podemos almacenar 64 bits, en el float y en el int solo 32 bits. Esto puede provocar la pérdida de información, para que este código funcione, podemos hacer el casting, colocar (float) delante de 3.14, pero lo haremos de otra manera.

7) Dile a Java que el valor 3.14 es float, por lo tanto, pon f al final del número, nuestro código se verá así:

```
public class TestConversion {

    public static void main(String[] args){

        float puntoFlotante = 3.14f;

        double salario = 1270.50;
        int valor = (int) salario;
        System.out.println(valor);
    }
}
```

```
double valor1 = 0.2;  
double valor2 = 0.1;  
double total = valor1 + valor2;  
  
System.out.println(total);  
}  
  
}
```

[COPIA EL CÓDIGO](#)

¡Ahora nuestro código se compila!